A Method for Correcting Image Distortion Perceived by a Dynamic Observer in Small Projection-Based Immersive Displays

Sanjeeb Nanda
Timothy Dunn
SDS International Inc., Advanced Technologies Division
3403 Technological Avenue, Suite 7
Orlando, FL 32817
407-282-4432
snanda@sdslink.com, tdunn@sdslink.com

Keywords:
Projection, distortion correction, immersive displays, dynamic viewpoint.

ABSTRACT: *The use of projection-based immersive displays in training continues to proliferate rapidly with continually increasing demands placed on its infrastructure and capabilities. In particular, it has been employed for military training with significant benefits. However, the constraints of cost, fabrication and portability sometimes require uniform downscaling of the dimensions of such displays with undesirable consequences. Images rendered on smaller display surfaces can appear distorted when the position of the observer's eye point and the projector are not coincident or when the observer's eye point is not fixed. In fact, the smaller the surface, the greater the distortion perceived in the projected image for any given displacement of the observer's eye point from the projector. This paper discusses a method to overcome this challenge by ensuring that the image seen by a dynamic observer is consistent regardless of the position of their eye point. The method interpolates commonly used projection surfaces from a finite set of screen coordinates and adjusts the projected image dynamically over that surface to provide a consistent field of view to the observer.*

1. Introduction

The application of projection-based virtual reality systems continues to gain popularity in numerous areas of science as well as entertainment. However, such systems remain economically unviable to many potential customers with limited budgets in spite of significant progress to make them more affordable [7], [8]. One of the steps taken to make these systems economically feasible is the increased assimilation of Common Off The Shelf (COTS) hardware to drive down production cost [1]. However, in spite of this, the cost of fabricating large projection surfaces is generally high. Furthermore, achieving sufficient brightness of images projected onto larger surfaces requires the use of more expensive high-end projectors or more elaborate configurations employing a greater number of them. Needless to say, this has a detrimental effect on cost as well as on the ease of deployment.

One potential approach to surmount the aforementioned problems is to scale down the size of the display surfaces. This can reduce the manufacturing cost of the systems, enhance the ease of setup and improve portability. While portable, single-projector immersive systems have proliferated widely, they too suffer from significant drawbacks. First, the fields of view they span are constrained to the maximum angle of projection that can be achieved using a single projector. Second, they are oblivious to the distortion in the projected images perceived by the observer when their head position changes with respect to the screen. As a result, elements in an image projected onto parts of the screen closer to the observer appear disproportionately larger than those on parts of the screen further away.

In this paper we present the methods to overcome the aforesaid challenges by creating an immersive projection system that can utilize a broad range of small-sized projection surface geometries, while ensuring that the images viewed by an observer is distortion-free regardless of their position within it. Distortion correction is realized by the application of simple vector algebra on the interpolated geometries of screens described by quadrics.

2. Interpolating a Screen

The class of algebraic surfaces called quadrics models the geometries of many commercially available projection screens. In fact, quadrics encompass 17 standard geometrical forms that include ellipsoids, elliptic cylinders, elliptic paraboloids, hyperbolic cylinders and parabolic cylinders, and have been widely used for modeling various 3D surfaces [2], [5]. Quadrics are defined by the second order equation, $ax^2 + by^2 + cz^2 + 2fxy + 2gxz + 2hyz + 2px + 2qy + 2rz + d = 0$ (Eq. 1). This equation may be succinctly expressed by the matrix-vector form,

$$p^T Ap = 0, \text{ where } p = \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \text{ and } A = \begin{bmatrix} a & f & g & p \\ f & b & h & q \\ g & h & c & r \\ p & q & r & d \end{bmatrix} \quad \text{(Eq.2)}$$

Thus, there is an isomorphism between A and a quadric defined by it. Therefore a surface that is assumed to be a quadric may be modeled by solving for the ten unknown variables constituting the entries of A. One technique for solving such an equation involves the following steps.

(i) Transform Eq. 2 to the form $p^T Bp = 0$ (Eq. 3), where B is a 4 x 4 matrix comprised of nine variables and the constant 1.
(ii) Construct a system of nine linear equations in nine variables a, b, c, f, g, h, p, q, and r using Eq. 2. Each equation is derived using the coordinates of a unique point $p_k = (x_k, y_k, z_k)$ on the surface of the projection screen, $k = 1, 2, \ldots 9$, satisfying the conditions, $x_k + y_k + z_k \neq 0$ and $x_k + y_k + z_k \neq -2$.
(iii) Use Gaussian elimination to solve for the nine variables, a, b, c, f, g, h, p, q, and r in the system of linear equations constructed in the preceding step.

Step (i): We transform Eq. 2 to the form $p^T Bp = 0$ by multiplying each entry of A by the common factor $1/d$. The need for this step will become apparent later in step (iii) of our discussion.

Step (ii): Next we construct a set of nine linear equations in nine variables a, b, c, f, g, h, p, q, and r, as follows. For each $i = 0, 1,\ldots, 8$, we choose a point (x_i, y_i, z_i) on the surface of the screen that satisfies the conditions, $x_i + y_i + z_i \neq 0$ and $x_i + y_i + z_i \neq -2$, and substitute the values x_i, y_i, and z_i into the variables x, y and z in Eq. 1 to yield an a system of equations in the nine aforesaid variables. The following matrix vector form represents the resulting system of equations.

$B\vec{x} = \vec{d}$ (Eq. 4), where

$$\vec{x} = \begin{bmatrix} \frac{a}{d} & \frac{b}{d} & \frac{c}{d} & \frac{f}{d} & \frac{g}{d} & \frac{h}{d} & \frac{p}{d} & \frac{q}{d} & \frac{r}{d} \end{bmatrix}^T,$$

$\vec{d} = \begin{bmatrix} -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 \end{bmatrix}$, and

$$B = \begin{bmatrix} b_{00} & b_{01} & \ldots & b_{08} \\ b_{10} & b_{11} & \ldots & b_{18} \\ \vdots & \vdots & & \vdots \\ b_{80} & b_{81} & \ldots & b_{88} \end{bmatrix}, \text{ with}$$

$b_{i0} = x_i^2$, $b_{i1} = y_i^2$, $b_{i2} = z_i^2$,
$b_{i3} = 2x_i y_i$, $b_{i4} = 2x_i z_i$, $b_{i5} = 2y_i z_i$,
$b_{i6} = 2x_i$, $b_{i7} = 2y_i$, $b_{i8} = 2z_i$,
for $i = 0,1,\ldots,8$.

If B is nonsingular, solving this system of linear equations using Gaussian elimination is straightforward. On the other hand, B is singular if and only if it is column-dependent, i.e., a linear combination of its columns equals the zero vector. Thus, for non-zero constants a_0, a_1, a_2, … a_8, we must have,

$$\sum_{k=0}^{8} a_k b_{ik} = 0, \text{ for } i = 0,1,\ldots,8. \quad \text{(Eq. 5)}$$

From Eq. 5 we obtain $a_i(x_i + y_i + z_i)(x_i + y_i + z_i + 2) = 0$, $i = 0, 1, \ldots, 8$. This is a contradiction since we chose $p_i = (x_i, y_i, z_i)$ such that, $x_i + y_i + z_i \neq 0$ and $x_i + y_i + z_i \neq -2$, for $i = 0, 1,\ldots, 8$. Therefore, B must be nonsingular and we must have a unique solution for Eq. 4.

We should note that, the two conditions imposed upon each $p_i = (x_i, y_i, z_i)$, namely, $x_i + y_i + z_i \neq 0$ and $x_i + y_i + z_i \neq -2$, are easily satisfied in practice by making the following realistic suppositions. We may assume that distance is measured in feet and that, the z-axis in a right-handed coordinate system intersects the screen at less than or equal to –2' and finally, the x and y coordinate values of each p_i are both negative.

Step (iii) Next we solve Eq. 4 using Gaussian elimination in the following manner. Let $B, \vec{d}$ and $\vec{x}$ be as described by Eq. 4. Let N be the number of rows (and also columns) of B, which equals 9. Then, the value of $\vec{x}$ satisfying Eq. 4 is obtained by invoking the following functions in the given order.

Triangulate (B, d, N);
BackSubstitute (B, d, x, N);

The function *Triangulate* transforms B into an upper triangular matrix and the function *BackSubstitute* then solves for $\vec{x}$ using B by back-substitution starting at the last row of B to yield the last coordinate of $\vec{x}$ and iteratively proceeding to the first row of B to yield the first coordinate of $\vec{x}$. Note that the following pseudo-code for the aforementioned functions describes the general technique for solving a system of linear equations with an arbitrary number of variables, and not just those restricted to the nine that are of interest to us in this case.

```
Triangulate (B, d, N)
{
    for i ← 0 to (N – 2) do
    {
        π ← Pivot (B, d, i, N);
        if ( | π | < ε ) then return false;

        for j ← i + 1 to (N – 1) do
        {
            ρ = B[j][i] / π;

            for k ← i + 1 to (N – 1) do
            { B[j][k] = B[j][k] – ρB[i][k]; }

            d[j] ← d[j] – ρd[i];
        }
    }
}

Pivot (B, d, i, N)
{
    j ← i; t ← 0;

    for k ← i to (N – 1) do
    {
        if (|B[k][i]| > t)
        { t ← |B[k][i]|;   j ← k; }
    }

    if (i < j) then
    { Swap rows B[i] and B[j]; Swap d[i] and d[j]; }

    return B[i][i];
}

BackSubstitute (B, d, x, N)
{
    for i ← N – 1 down to 0 do
    {
        x[i] ← (d[i] – DotProduct (B[i], x, i + 1,
                                   N – 1))/B[i][i];
    }
}

DotProduct (u, v, k1, k2)
{
    sum ← 0;

    for i ← k1 to k2 do
    { sum ← sum + u[i]v[i]; }

    return sum;
}
```

Upon solving for $\vec{x}$, we obtain the unique values for the nine components $\frac{a}{d}, \frac{b}{d}, \frac{c}{d}, \frac{f}{d}, \frac{g}{d}, \frac{h}{d}, \frac{p}{d}, \frac{q}{d}, \frac{r}{d}$ that specify the quadric representing the projection surface.

3. Preserving the Field of View

The distortion observed in an image viewed from positions not coinciding with the focal point of the projector is caused by the variance in the measure of either or both the horizontal and vertical field of views occupied by that image. Consider the example illustrated by Fig. 1. It shows two positions $p_1 = (x_1, y_1, z_1)$ and $p_2 = (x_2, y_2, z_2)$ from which an observer views an image occupying horizontal field of views α_1 and α_2 respectively on a screen viewed top-down. The dark arc overlaid atop the screen indicates the position of the image. Since α_2 is smaller than α_1 the image appears laterally compressed to the observer from position p_2 in comparison to that in position p_1. In contrast, Fig. 2 shows two positions q_1 and q_2 from which an observer views duplicate images occupying identical horizontal field of views β on a screen. Two dark overlapping arcs overlaid atop the screen indicate the respective positions of the images. Since the field of views encompassed by the images from positions q_1 and q_2 are identical, the images from those two positions appear undistorted to the observer relative to each other.

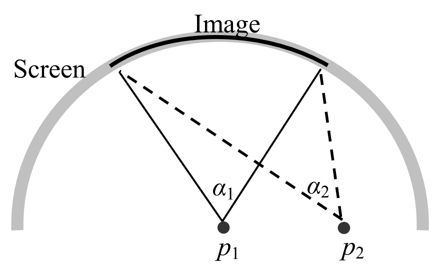

Fig. 1. Lateral compression observed by an observer when the field of view is narrower.

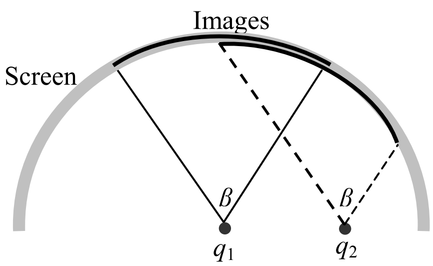

Fig. 2. Correcting distortion by preserving the field of view encompassed by an image.

4. Texture Projection

Consider an image defined by a texture that is projected onto a screen using m x n vertices. Suppose the observer's eye point is at position p_1 and each vertex (i, j) of the texture is mapped to the point $= p_1 + t' \vec{v}_{i,j}$ on the screen, where $\vec{v}_{i,j}$ is a normalized vector, $0 \le i < m$, $0 \le j < n$, and

t' a real. Now, suppose the position of the observer changes from point p_1 to p_2. Then, the field of view between any two image points to which vertices of the texture are mapped from p_1 can be preserved at p_2 by mapping each vertex (i, j) of the texture to the point $p_{i,j}$ on the screen, such that $p_{i,j} = p_2 + t\vec{v}_{i,j}$, for some $t > 0$. Since $p_{i,j}$ is a point on the surface of the screen, it must satisfy Eq. 3, i.e., $p_{i,j}{}^T B\, p_{i,j} = 0$. This is,

$$(p_2 + t\vec{v}_{i,j})^{\mathrm{T}} B\, (p_2 + t\vec{v}_{i,j}) = 0$$

$$\Rightarrow t^2 \vec{v}_{i,j}^{\mathrm{T}} B \vec{v}_{i,j} + t(\vec{v}_{i,j}^{\mathrm{T}} B p_2 + p_2^{\mathrm{T}} B \vec{v}_{i,j}) + p_2^{\mathrm{T}} B p_2 = 0$$

This is a quadratic in t that admits the two roots,

$$t = \frac{-(\vec{v}_{i,j}^{\mathrm{T}} B p_2 + p_2^{\mathrm{T}} B \vec{v}_{i,j}) \pm \sqrt{(\vec{v}_{i,j}^{\mathrm{T}} B p_2 + p_2^{\mathrm{T}} B \vec{v}_{i,j})^2 - 4(\vec{v}_{i,j}^{\mathrm{T}} B \vec{v}_{i,j})(p_2^{\mathrm{T}} B p_2)}}{2(\vec{v}_{i,j}^{\mathrm{T}} B \vec{v}_{i,j})}$$

Since B, p_2 and $\vec{v}_{i,j}$ are known, we can derive t and use it to obtain each $p_{i,j}$. Fig. 3 illustrates a hypothetical projection configuration consisting of a spherical screen illuminated by four projectors. Each projector is driven by a unique image generator, with their respective field of views controlled by an auto alignment module. The vectors $p_1 + t'\vec{v}_{i,j}$ and $p_2 + t\vec{v}_{i,j}$ are used for mapping vertex (i, j) of the texture from the observer's positions p_1 and p_2 respectively.

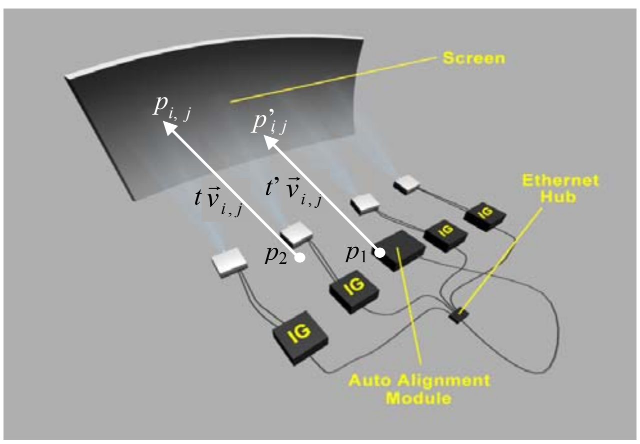

Fig. 3. A projection environment displaying the vectors for texture mapping.

The results of using this technique for correcting distortion are illustrated using the following three figures. First, Fig. 4 displays an image projected onto a cylindrical surface that is viewed by an observer whose eye point is coincident with that of the projector. Next, Fig. 5 displays the same image viewed by the observer whose eye point has been translated along the x-axis by some finite positive value. Note that the image and the grids of the screen upon which it is projected are distorted with components closest to the observer appearing disproportionately large. Finally, Fig. 6 displays the image corrected from the observer's new position and thus appearing identical to that viewed by the observer from the position in Fig. 4, while the grids of the screen upon which it is projected are distorted.

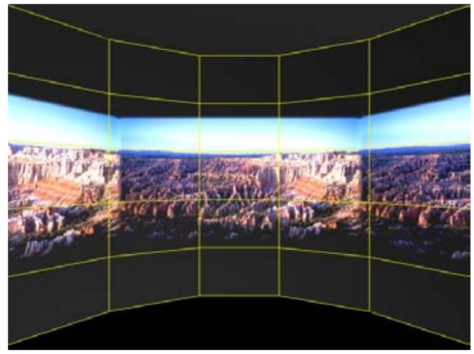

Fig. 4. Image viewed by observer from center position.

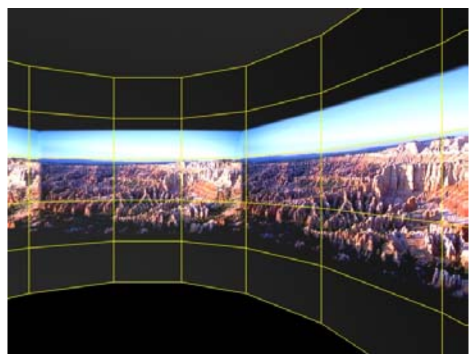

Fig. 5. Uncompensated view from offset position.

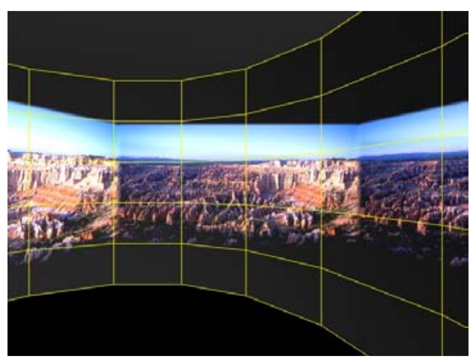

Fig. 6. Compensated view from offset position.

While the illustrations in Fig. 4, 5, and 6 demonstrate the previously described technique for correcting image distortion on a quadric surface that is cylindrical, the applicability of the method extends to all quadrics as stated earlier.

5. Conclusions

While the technique presented in this paper has immense value for correcting distortion, one of its drawbacks is that, distances perceived by the observer to objects in the image remain invariant regardless of the position of the observer. When objects are at sufficiently larger distances from the observer, the change in distance to an object corresponding to a change in the position of the observer is small enough compared to the distance to that object to be visually imperceptible. However, when an object is relatively close to an observer, a change in the position of the observer can be expected to produce a perceptible change in the distance to the object. Therefore, such methods for correcting image distortion should be applied only in those scenarios where occurrences of well-defined objects close to the observer are limited. For example consider the nose cone of an aircraft viewed by its pilot. One would expect a change in the perspective view of the cone in the event of a change in the position of the pilot's eye point. However, the application of such methods for correcting image distortion will result in the nose cone's appearance remaining invariant to the observer.

6. References

[1] R. Belleman, B. Stolk, R. de Vries, Immersive Virtual Reality on Commodity Hardware, *Proceedings of the 7th Annual Conference of the Advanced School of Computing and Imaging*, 2001, pp. 297--304.

[2] I. Douros and B. Buxton, "Three-Dimensional Surface Curvature Estimation using Quadric Surface Patches", *Scanning* 2002 *Proceedings*, Paris, May 2002.

[3] J. D. Foley, A. van Dam, S. Feiner, and J. Hughes, "Computer graphics: Principles and practice in C", Addison Wesley, 1995.

[4] C. Cruz-Neira, C. Sandin and T. DeFanti. "Surround-Screen Projection-Based Virtual Reality: The Design and Implementation of the CAVE," *Siggraph*, 1993, pp.135-142.

[5] K. Johnson, P. Smith and M. Abidi, "A Quadric Surface Projection Model for Wide Angle Lenses," *Proceedings of SPIE Intelligent Robots and Computer Vision* XVII, 1998, vol. 3522, pp. 424 – 434.

[6] P. Lyon, "Edge-blending Multiple Projection Displays On A Dome Surface To Form Continuous Wide Angle Fields-of-View," Proceedings of 7th I/ITEC, 1985, pp. 203 – 209.

[7] D. Pape, J. Anstey, and G. Dawe. A Low-Cost Projection Based Virtual Reality Display. The Engineering Reality of Virtual Reality 2002, *SPIE Stereoscopic Displays and Virtual Reality Systems* IX, 2002, vol. 4660, pp. 483 – 491.

[8] J. Pair, C. Jensen, J. Flores, J. Wilson, L. Hodges, and D. Gotz, "The nave: Design and implementation of non-expensive immersive virtual environment," *Siggraph 2000 Sketches and Applications*, pp. 238, 2000.

[9] R. Raskar, G. Welch, M. Cutts, A. Lake, L. Stesin, and H. Fuchs. "The Office of the Future: A Unified Approach to Image-Based Modeling and Spatially Immersive Displays," *International Conference on Computer Graphics and Interactive Techniques*, 1998, pp. 179 – 188.

[10] P. J. Schneider and D. H. Eberly, *Geometric Tools for Computer Graphics*, Morgan Kaufmann, 2003.

7. Author Biographies

Mr. Sanjeeb Nanda is a research and development engineer with the Advanced Technologies Division of SDS International. His experience spans the areas of simulation, biometrics, parallel computing, and fault-tolerance in computing. He is currently pursuing a doctorate in Computer Science.

Mr. Timothy Dunn is a visual systems engineer with the Advanced Technologies Division of SDS International. His experience spans the areas of modeling and simulation.